

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-181598

(43)Date of publication of application : 30.08.2000

(51)Int.Cl.

G06F 3/00

(21)Application number : 11-349681

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 09.12.1999

(72)Inventor : DAVID JAMES HETHERINGTON
DAVID BRUCE KUMATA

(30)Priority

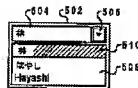
Priority number : 98 211810 Priority date : 15.12.1998 Priority country : US

(54) METHOD AND DEVICE FOR CHANGING DISPLAY OF MULTI FIELD CHARACTER STRING AND COMPUTER PROGRAM PRODUCT

(57)Abstract:

PROBLEM TO BE SOLVED: To change the user interface display of a character string by providing a user control for selectively displaying/editing field contents in a multi field character string object having separate fields respectively including different representations of the character string.

SOLUTION: When this device is started by a user's pointing device, etc., a selection controlling part 506 generates a drop down box 508 including all three IString objects in the order of baseString, sortString and altString. Then, three character strings are shown as a list with sequences in the box 508. When a user clicks a desired entry in the drop down box 508, it is highlighted by a bar 510. Thus, it is possible to select display in a different field.



LEGAL STATUS

[Date of request for examination] 09.12.1999

[Date of sending the examiner's decision of rejection] 04.06.2003

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3841991

[Date of registration] 18.08.2006

[Number of appeal against examiner's decision of rejection] 2003-016898

[Date of requesting appeal against examiner's decision of rejection] 02.09.2003

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-181598

(P2000-181598A)

(43) 公開日 平成12年6月30日 (2000.6.30)

(51) Int.Cl.¹

G 0 6 F 3/00

識別記号

6 5 4

F I

G 0 6 F 3/00

テグコード (参考)

6 5 4 C

審査請求 有 請求項の数24 O L (全 25 頁)

(21) 出願番号 特願平11-349681

(22) 出願日 平成11年12月9日 (1999.12.9)

(31) 優先権主張番号 0 9 / 2 1 1 8 1 0

(32) 優先日 平成10年12月15日 (1998.12.15)

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーションINTERNATIONAL BUSIN
ESS MACHINES CORPO
RATIONアメリカ合衆国10504、ニューヨーク州
アーモンク (藩地なし)

(74) 代理人 100086243

弁理士 坂口 博 (外1名)

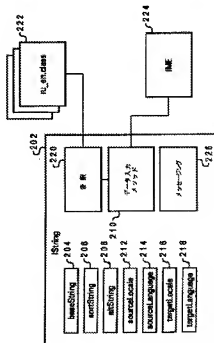
最終頁に続く

(54) 【発明の名称】 マルチフィールド文字列の表示を変更する方法、装置およびコンピュータ・プログラム製品

(57) 【要約】

【課題】 文字列のユーザ・インタフェース表示を変更する方法、システムおよびコンピュータ・プログラム製品を提供する。

【解決手段】 文字列の異なる表現 (たとえば漢字記号、漢字記号を発音どおりにつづる表音文字および漢字記号を発音どおりにつづるローマ字) をそれぞれを含む別個のフィールドを有するマルチフィールド文字列オブジェクト内のフィールドのコンテンツを選択的に表示および/または編集するためのユーザ制御を提供する。したがって、種々のフィールドは、一つの文字列の識別名、意味および発音情報をカプセル化する。ユーザ制御装置とマルチフィールド文字列オブジェクトとの間の制御インタフェースが、ユーザ制御内のメソッドをマルチフィールド文字列オブジェクトに対して実行することを可能にする。マルチフィールド文字列オブジェクトに現在選択されている表示フィールドのコンテンツが選択制御とともに表示される。



【特許請求の範囲】

【請求項1】 マルチフィールド文字列の表示を変更するための方法であって、

ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するステップと、前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するステップと、

前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するステップと、

前記選択制御のユーザ起動に responding、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するステップと、を含むことを特徴とする方法。

【請求項2】 ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供する前記ステップが、

前記マルチフィールド文字列オブジェクトの第一のフィールド内で第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するステップと、前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するステップと、をさらに含む請求項1記載の方法。

【請求項3】 前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示する前記ステップが、

前記マルチフィールド文字列オブジェクトのどのフィールドか前記現表示フィールドであるかを識別するステップと、

識別した前記フィールドから文字列を検索するステップと、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するステップと、をさらに含む請求項1記載の方法。

【請求項4】 前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択に responding、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するステップをさらに含む請求項1記載の方法。

【請求項5】 前記新たな現表示フィールドから文字列を検索するステップと、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するステップとをさらに含む請求項4記載の方法。

【請求項6】 前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するステップと、

前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するステップと、

前記選択制御のユーザ起動に responding、前記現表示フィールドに入力するための新たな文字列を受けるステップと、

前記新たな文字列を前記マルチフィールド文字列オブジェクト内の前記現表示フィールド中に記憶するステップとをさらに含む請求項1記載の方法。

【請求項7】 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示する前記ステップが、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するステップと、

前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するステップとをさらに含む請求項1記載の方法。

【請求項8】 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示する前記ステップが、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するステップと、

前記文字列を発音どおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するステップと、

前記文字列を発音どおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するステップとをさらに含む請求項1記載の方法。

【請求項9】 マルチフィールド文字列の表示を変更するためのシステムであって、

ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための手段と、

前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための手段と、

前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するための手段と、

前記選択制御のユーザ起動にตอบสนองして、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための手段と、

を含むことを特徴とするシステム。

【請求項10】 ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための前記手段が、

前記マルチフィールド文字列オブジェクトの第一のフィールド内で第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するための手段と、

前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するための手段と、をさらに含む請求項9記載のシステム。

【請求項11】 前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための前記手段が、

前記マルチフィールド文字列オブジェクトのどのフィールドが前記現表示フィールドであるかを識別するための手段と、

識別した前記フィールドから文字列を検索するための手段と、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための手段と、をさらに含む請求項9記載のシステム。

【請求項12】 前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択にตอบสนองして、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するための手段をさらに含む請求項9記載のシステム。

【請求項13】 前記新たな現表示フィールドから文字列を検索するための手段と、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための手段とをさらに含む請求項12記載のシステム。

【請求項14】 前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するための手段と、前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するための手段と、

前記選択制御のユーザ起動にตอบสนองして、前記現表示フィールドに入力するための新たな文字列を受け取るための手段と、前記新たな文字列を前記マルチフィールド文字列オブ

ジェクト内の前記現表示フィールド中に記憶するための手段とをさらに含む請求項9記載のシステム。

【請求項15】 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記手段が、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための手段と、

前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための手段とをさらに含む請求項9記載のシステム。

【請求項16】 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記手段が、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための手段と、

前記文字列を発音どおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための手段と、

前記文字列を発音どおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するための手段とをさらに含む請求項9記載のシステム。

【請求項17】 マルチフィールド文字列の表示を変更するためのコンピュータ使用可能な媒体内のコンピュータ・プログラム製品であって、ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための命令と、

前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための命令と、

前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するための命令と、

前記選択制御のユーザ起動にตอบสนองして、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための命令と、を含むことを特徴とするコンピュータ・プログラム製品。

【請求項18】 ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供す

るための前記命令が、
前記マルチフィールド文字列オブジェクトの第一のフィールド内で第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するための命令と、

前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するための命令と、をさらに含む請求項17記載のコンピュータ・プログラム製品。

【請求項19】前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための前記命令が、

前記マルチフィールド文字列オブジェクトのどのフィールドが前記現表示フィールドであるかを識別するための命令と、

識別した前記フィールドから文字列を検索するための命令と、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための命令と、をさらに含む請求項17記載のコンピュータ・プログラム製品。

【請求項20】前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択にตอบสนองして、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するための命令をさらに含む請求項17記載のコンピュータ・プログラム製品。

【請求項21】前記新たな現表示フィールドから文字列を検索するための命令と、

検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための命令とをさらに含む請求項20記載のコンピュータ・プログラム製品。

【請求項22】前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するための命令と、前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するための命令と、

前記選択制御のユーザ起動にตอบสนองして、前記現表示フィールドに入力するための新たな文字列を受け取るための命令と、

前記新たな文字列を前記マルチフィールド文字列オブジェクト内の前記現表示フィールド中に記憶するための命令とをさらに含む請求項17記載のコンピュータ・プログラム製品。

【請求項23】前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記命令が、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための命令と、

前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための命令とをさらに含む請求項17記載のコンピュータ・プログラム製品。

10 【請求項24】前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記命令が、

前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための命令と、

前記文字列を発音どおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための命令と、

20 前記文字列を発音どおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するための命令とをさらに含む請求項17記載のコンピュータ・プログラム製品。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は一般に、文字列のユーザ・インタフェース表示の変更に関し、特に、マルチフィールド文字列オブジェクトのユーザ・インタフェース表示の変更に関する。さらに具体的には、本発明は、ユーザ・インタフェース表示内でのマルチフィールド文字列オブジェクトからのフィールドの選択的な表示に関する。

【0002】

【従来の技術】多国籍企業は、地球上に分散した多数の地域に及ぶ情報システム（IS）ネットワークを運用することが多い。そのようなネットワークの有用性を最大限にするため、各国内での運用は、その国の現地語で実施される傾向にある。可能ならば、ユーザ・アプリケーションにおける抽象オブジェクトの名称は現地語であり、抽象オブジェクトが表す現地語の機能、都市または人名に適合する。システム管理ソフトウェアの場合、しばしば抽象オブジェクトは、全世界的企業の現地オフィスそれぞれを表す。

【0003】抽象オブジェクトの名称が現地語および現地語の基礎にある文字セットを使用する場合、そのような全世界的ネットワークの集中管理は困難であるか、不可能である。エジプトにあるオフィスの場合、抽象オブジェクトはしごく当然にアラビア語で命名されるだろう

し、ロシアにあるオフィスの場合、キリル文字セットを使用してオブジェクトを命名するだろうし、日本にあるオフィスの場合、オブジェクトは日本語で命名されるだろう。しかし、企業の本部の1Sスタッフがこれらのオブジェクトを調査しようすると、問題が生じる。米国にある多国籍企業本部の1Sスタッフは、アラビア語や日本語を読める可能性は低く、キリル文字でさえ理解する可能性は低い。

【0004】たとえば日本語は、単なる音を表すアルファベットを有する、その代わり、単なる音ではなく概念およびオブジェクトに対応する記号（「表意文字」）を有する非常に大きな文字セットを有する表意文字言語である。たとえば、1981年に採用された日本語の常用漢字表は1945個の記号を含む。漢字になじみのないユーザにとって、日本語で命名された特定の抽象オブジェクトを識別することは困難であり、また、そのような抽象オブジェクトに関して英語および日本語を話す相手と電話で話し合うことさえ困難である。

【0005】さらには、日本語では、同じ文字が多数の意味または発音をもつことがあるため、表意文字を見るだけでは、正しい意味または発音に関して手がかりを得ることはできない。たとえば、図9に示す文字は「Wes t」または「Spain」のいずれを表すこともでき、図10に示す記号は「hayashi」または「rin」（もしくは「lin」）のいずれにも発音することができ、図11に示す文字は「suga no」、「suga ya」、「kan no」または「kan ya」のいずれにも発音することができる。この状況は、一部には、漢字を中国語から採用した日本語の歴史に基づく。したがって、たとえば、図10に示す「rin」記号は音読み、すなわち、基本的にはこの文字が日本に輸入された当時の中国語発音を模したものであり、一方「hayashi」は別読み、すなわち、同じ意味を有する、この文字に割り当てられた日本語である。

【0006】したがって、ユーザ・インタフェースにおける文字列の表示言語を選択的に制御するための機構を提供することが望ましい。さらに、その機構が、ユーザがユーザ・インタフェースの表示言語を変更し、表示フィールドのコンテンツを編集することを許すならば、有利である。

【0007】

【発明が解決しようとする課題】したがって、本発明の一つの目的は、文字列のユーザ・インタフェース表示を変更する方法、システムおよびコンピュータ・プログラム製品を提供することである。

【0008】本発明のもう一つの目的は、マルチフィールド文字列オブジェクトのユーザ・インタフェース表示を変更する方法、システムおよびコンピュータ・プログラム製品を提供することである。

【0009】本発明のさらに別の目的は、ユーザ・インタフェース表示内のマルチフィールド文字列オブジェ

クトからのフィールドを選択的に表示する方法、システムおよびコンピュータ・プログラム製品を提供することである。

【0010】

【課題を解決するための手段】前記目的は、以下に記載するようにして達成される。文字列の異なる表現（たとえば漢字記号、漢字記号を発音どおりにつづる表音文字および漢字記号を発音どおりにつづるローマ字）をそれぞれを含む別個のフィールドを有するマルチフィールド文字列オブジェクト内のフィールドのコンテンツを選択的に表示および/または編集するためのユーザ制御を提供する。したがって、種々のフィールドは、一つの文字列の識別名、意味および発音情報をカプセル化する。ユーザ制御とマルチフィールド文字列オブジェクトとの間の制御インタフェースが、ユーザ制御内のメソッドをマルチフィールド文字列オブジェクトに対して実行することを可能にする。マルチフィールド文字列オブジェクトに現在選択されている表示フィールドのコンテンツが選択制御とともに表示される。選択制御を起動すると、マルチフィールド文字列オブジェクト内の全フィールドのコンテンツの順序付け一覧を表示することができ、表示された一覧からフィールドを選択することによって現在表示フィールドを変更することができる。したがって、文字列の一つの表現に基づいて文字列を認識、理解または発音することができないユーザは、ビューを切り換えて、同じ文字列の、ユーザにとってなじみのある異なる表現を見ることができ、また、現在選択されている表示フィールドのコンテンツを編集することもできる。

【0011】以下の詳細な説明により、本発明の上記ならびにさらなる目的、特徴および利点が明らかになる。

【0012】

【発明の実施の形態】ここで図面、特に図1を参照すると、本発明の好ましい実施態様を実現することができるデータ処理システムのブロック図が示されている。データ処理システム100は、たとえば、International Business Machines社（ニューヨーク州Armonk）から市販されているパーソナル・コンピュータのAptiva（登録商標）の1機種であることができる。データ処理システム100はプロセッサ102を含み、例示する実施態様では、このプロセッサがレベル2（L2）キャッシュ104に接続され、このキャッシュが逆にシステム・バス106に接続されている。例示する実施態様では、データ処理システム100は、表示装置120のためのユーザ・インタフェース情報を受けるシステム・バス106に接続されたグラフィックス・アダプタ118を含む。【0013】同じくシステム・バス106には、システム・メモリ108および出力（I/O）バス・ブリッジ110が接続されている。I/Oバス・ブリッジ110は、I/Oバス112をシステム・バス106に結合して、一方のバスから他方のバスへのデータ・トランザ

クシヨンを中継および／または変換する。周辺装置、たとえばハードディスク・ドライブであってもよい揮発性記憶装置114および従来のマウス、トラックボールなどを含むキーボード／ポインティング・デバイス116が1/0バス112に接続されている。

【0014】図1に示す例示的な実施態様は、本発明を説明する目的のみに提示したものであり、当業者は、形態および機能の両方において多数の変形が可能であることを認識するであろう。たとえば、データ処理装置100はまた、コンパクト・ディスク読み取り専用メモリ(CD-ROM)もしくはデジタル・ビデオ・ディスク(DVD)ドライブ、サウンド・カードおよびオーディオ・スピーカならびに他多数のオプション部品を含むかもしれない。そのような変形はすべて本発明の本質および範囲に入るものと考えられる。データ処理システム100および以下に示すJava実例は、説明のための例としてのみ提供するものであり、機器構成の限定を暗示するものではない。当業者は、利用することができる多数のプログラム言語を認識するであろう。それらはすべて本発明の本質および範囲に包含されると考えられる。

【0015】図2を参照すると、本発明の好ましい実施態様にしたがって、ユーザ・インタフェースに代替表示言語選択を提供するために使用されるマルチフィールド文字列クラスの図が示されている。データを複数の人間言語で表示しなければならない国際的なコンピューティング環境における根本的な問題は、発話される語が一般に情報、たとえばその語の意味を通じて、文脈からおおよ／または語尾変化から多数の態様または属性でカプセル化することである。語は、データ処理システムにおける操作または表示のために視覚的または電子的表現に簡約されると、いくつかの属性および連想される意味の多くを失うおそれがある。データ処理システムにとってもっとも重大なことに、語の視覚的表現は、その語の正しい綴りもしくは発音または指定されたソート順序での語の正しい配置に関して手がかりを与えることができない。この問題に対処するためには、国際文字列(「Istr

```

/**The base text String*/
protected String baseString;
/**The related text String for proper collation*/
protected String sortString;
/**The related alternate text String (pronunciation key)*/
protected String altString;
/**The source locale, as an ISO-3166 code: used for collation*/
protected String sourceLocale;
/**The source language, as an ISO-639 code*/
protected String sourceLanguage;
/**The source variant defined for EBCDIC and case mapping*/
protected String sourceVariant;
/**The target locale, as an ISO-3166 code*/
protected String targetLocale;

```

ing)」クラス202を用いることができる。

【0016】IStringクラス202は、好ましくは、類似メソッドを含めることによってJava文字列クラスのように挙動する、Java文字列クラスに類似したJavaクラスである。文字列クラスの元の挙動の大部分が保存され、さらなる機能が追加され、必要な場合にのみ使用されるはずである。IStringクラス202は、発話される語の意味のうち、その語が視覚的表現に簡約された場合に通常は失われる部分を捕らえるデータ型である。IStringクラス202は、好ましくは、システム内のすべてのオブジェクト名およびシステム・メッセージに使用される。

【0017】IStringクラス202構造は、名前、メッセージ、データまたは文字列オブジェクトごとに3個の異なる文字列、すなわちbaseString204、sortString206およびaltString208を含む。baseString204は、ユーザ・インタフェース表示装置中でデフォルトによって用いられるIString202内の文字列であり、普通はIStringオブジェクトが生成される現地語でユーザによって入力される元の文字列を含むことができる、sortString206もまた、文字列であることができ、非音声的言語およびbaseString204の2進値だけに基づいてソートするのが困難な言語の正しいソートを可能にするために用いられる。altString208は、いかなる文字列であることもできるが、従来は、baseString204に含まれるデータの発音のローマ字セット表現で満たされているべきである。したがって、IStringクラス202は、オブジェクト名、システム・メッセージおよび他のデータの元の元の文字列(baseString204)、ソート・キー(sortString206)および発音キー(altString208)を含む。

【0018】Javaで実現されると、IStringクラス202オブジェクトのコンストラクタは、以下のフィールドからなることができる。

【0019】

```

11
/**The target language, as an ISO-639 code*/
protected String targetLanguage;
/**The target variant defined for EBCDIC and case mapping*/
protected String targetVariant;

```

【0020】ISO-3166によって定義される大文字2文字のISO国コードおよびISO-639によって定義される小文字2文字のISO言語コードは、インターネット上の多様なソースから用意に入手することができる。

【0021】表1は、IStringデータ型202内のデータがテーブルとして表現されたときにどのように見えるかを示す。

【0022】

フィールド	型	データ
baseString	Java文字列	ユーザの文字列
sortString	Java文字列	言語/地域依存
altString	Java文字列	言語/地域依存
sourceLocale	Java文字列	ISO-3166コード、たとえばUS
sourceLanguage	Java文字列	ISO-639コード、たとえばen
sourceVariant	Java文字列	可変コード
targetLocale	Java文字列	ISO-3166コード、たとえばJP
targetLanguage	Java文字列	ISO-639コード、たとえばja
targetVariant	Java文字列	可変コード

表1

【0023】コンテンツが言語または地域から独立して

※る。

いる新たな空のIStringクラス・オブジェクト202のJ

【0024】

avaコンストラクタは、以下のものであることができ ※

```

/*****
 *
 * <P></P>
 *
 * <dt><b>Description:</b><dd>
 * <p>Allocate a new IString containing no characters in the default
 * locale. </p>
 *
 *****/
public IString() {
    this.baseString=new String();
    this.sortString=new String();
    this.altString=new String();
    init();
}

```

【0025】IStringクラス202データ型のオブジェクトをオブジェクト・データベース(OJB)に記憶し、ただし、共通オブジェクト要求プロカー・アーキテクチャ(CORBA)アプリケーションによるIString

データの操作を可能にするため、インタフェース定義言語(IDL)クラスが次のように定義されるべきである。

【0026】

```

ストラクツIString{
    文字列baseString; //ベース・テスト文字列
    文字列sortString; //照合のための関連の文字列
    文字列altString; //関連の代替文字列(発音)
    文字列sourceLocale; //ISO-3166コードとしてのソース地域
    文字列sourceLanguage; //ISO-639コードとしてのソース言語
    文字列sourceVariant; //ソース可変コード
    文字列targetLocale; //ISO-3166コードとしての標的地域
}

```



```

13
文字列targetLanguage: //ISO-639コードとしての標準言語
文字列targetVariant://標的の変種コード
}

```

【0027】baseString204、sortString206およびaltString208のコンテンツは、IStringクラス202内のデータ入力メソッド210によって入力されるユニコード文字列であることが好ましいが、必ずしもそうである必要はない。データ入力メソッド210ならびにbaseString204、sortString206およびaltString208のコンテンツは、少なくとも一部には、sourceLocaleフィールド212、sourceLanguageフィールド214、targetLocaleフィールド216および言語およびtargetLanguageフィールド218によって定義される言語および地域パラメータに依存する。

【0028】データ入力メソッド210は、基礎にある*

```

/*****
 *
 * <P></P>
 *
 * <dt><b>Description:</b><dd>
 * <p>Allocate a new IString containing no characters in the
 * specified locale. </p>
 *
 *****/
public IString(Locale loc) {
    this.baseString=new String();
    this.sortString=new String();
    this.altString=new String();
    this.sourceLocale=loc.getLocale();
    this.sourceLanguage=loc.getLanguage();
    init();
}

```

【0030】IStringクラス202オブジェクトへのデータの入力は、好ましくは、地域依存性または言語依存性である。sourceLanguage属性214およびtargetLanguage属性218が、データ入力メソッド210によってデータがIStringクラス202オブジェクトに入力される方法を制御する。sourceLanguage属性214は、IStringクラス・オブジェクトが生成されるホスト・システムの言語属性にセットすることもできる。targetLanguage属性218は、その言語にセットしてもよいし、あるいはまた、共通の「全世界的」言語、たとえば英語にセットしてもよい。データ入力メソッド210は、sourceLanguage属性214とtargetLanguage属性218とを比較して、IStringクラス・オブジェクト202の中で何をbaseString204、sortString206およびaltString208に入力するかを決定する。

【0031】文字列がIStringクラス202のデータ入力メソッド210によってbaseString204、sortString206およびaltString208のフィールドに入力さ

*ホスト・システムによって用いられる地域および/または言語に依存するため、新たなIStringオブジェクト202の生成は、好ましくは、IStringオブジェクト202が生成されるホスト・システムの地域および言語特性をsourceLocaleフィールド212およびsourceLanguageフィールド214に配属する結果となる。IStringクラス・オブジェクト202が生成されるホスト・システムから決定された指定地域および言語の新たな空のIStringを割り当てるためのコンストラクタは、以下のものであることができる。

【0029】

れる。このメソッドは、ユーザの直接入力もしくは仕様からのデータ、音訳エンジン220からのデータまたは入力メソッド・エディタ（IME）224からのデータを選択的に利用することができる。targetLanguage属性218がデフォルトとして英語にセットされている場合には、データ入力メソッド210は、データがユーザによって入力される際の言語（sourceLanguage属性214）によって用いられる文字セットに基づいてbaseString204、sortString206およびaltString208のフィールドのコンテンツを決定する。

【0032】ローマ字セットを使用する言語の場合、ユーザ入力は、データ入力メソッド210により、IStringクラス202の3フィールド（baseString204、sortString206およびaltString208）すべてに配置される。適当なコンストラクタは、以下のものであることができる。

【0033】

```
/* **** */
```

```
*
* <P></P>
*
* <dt><b>Description:</b><dd>
* <p>Allocate a new String which contains the same sequence of
* characters as the string argument in the specified locale.</p>
*
```

```
/* **** */
```

```
public String(String str, Locale loc){
    this.baseString=new String(str);
    this.sortString=new String(str);
    this.altString=new String(str);
    this.sourceLocale=loc.getLocale();
    this.sourceLanguage=loc.getLanguage();
    init();
}
```

【0034】大部分の地域および言語の場合、入力される文字列は、`String`オブジェクト202の3フィールドすべてに入力される。`targetLanguage`属性218が英語にセットされていないならば、`sourceLanguage`属性214および`targetLanguage`属性218で識別される言語が共通の文字セットを使用する（たとえばスペイン語とアプリケーションの場合のように両方ともローマ字を使用する）ときには、データ入力メソッド224は、ユーザー入力文字列を3フィールドすべてに入力することになる。

【0035】表IIは、ホスト言語および地域がローマ字セットを使用する場合にデータを`String`クラス202フィールドに入力する方法を示す。

【0036】

フィールド	型	データ
<code>baseString</code>	Java文字列	Hetherington
<code>sortString</code>	Java文字列	Hetherington
<code>altString</code>	Java文字列	Hetherington
<code>sourceLocale</code>	Java文字列	US
<code>sourceLanguage</code>	Java文字列	en
<code>targetLocale</code>	Java文字列	US
<code>targetLanguage</code>	Java文字列	en

表II

【0037】望むならば、フィールドを個々に編集し、ソート目的のために、より低いソート値（たとえば「AA A. Hetherington」）を有する文字列を`sortString`206に挿入することによってオブジェクトを人工的にプロモート（artificial promotion）してもよい。

【0038】ローマ字セットを使用しないが、ローマ字

```
/* **** */
```

```
*
* <P></P>
*
```

セットに音声マッピングすることができる文字セットを使用する言語の場合、ユーザー入力は、データ入力メソッド210により、`baseString`204および`sortString`206に入力されるが、その入力の音訳された発音どおりの表現は`altString`208に配置される。音訳エンジン220内の内部メソッドを使用して、パスされた文字列を`altString`208の発音どおりのローマ字表現に音声マッピングして、入力された文字を、元の言語の文字セットになじみのない人々にとって理解できる他の文字に音訳する。

【0039】`altString`208のコンテンツを生成するため、音訳エンジン220は、マッピングテーブルを含む適当なJavaリソース・ファイル222を選択して、`altString`208に配置すべき代替文字列を生成する。使用される具体的なリソース・ファイルの選択は、ソース言語と標的言語との組み合わせに依存する。Javaリソース・ファイル222は、マッピングが実行される言語の組み合わせのために命名される。図2に示す実例例では、`ru_en.class`が、ロシア語（キリール文字）を英語（ローマ字）にマッピングするためのものである。リソース・ファイル222の構造は、外語文字および対応するローマ字のための関連のエントリを有するテーブルである。

【0040】パスされた文字列から`altString`208が音訳される`String`オブジェクトに適当なコンストラクタは、以下のものであることができる。

【0041】

```

17
*<dt><b>Description:</b><dd>
*<p>Allocate a new IString. The baseString and sortString are the
*passed string, the altString is transliterated into the target
*language.</p>
*
*****
public IString(String str){
    this.baseString=new String(str);
    this.sortString=new String(str);
    if(isSameLanguage()){
        this.altString=new String(str);
    }
    else
        this.altString=transmogrify(str,
                                this.sourceLanguage,
                                this.targetLanguage);
}

```

【0042】‘変形’メソッドは、上述した音訳エンジン220内の内部メソッドである。入力された文字が音訳されて得られる文字セットは、例示する実施態様では英語にセットされているとみなされるtargetLanguage属性218から決定される。しかし、適当なりソース・ファイル222を与えられ、文字は、いかなる二言語間でも音訳することができ、そのために、一方の言語の文字が他方の言語の1個以上の文字に音声マッピングさ

される。

【0043】表IIIは、言語が、ローマ字セットに対してマッピングする非ローマ字セット、たとえばロシア語のキリール文字を使用する場合にデータ入力メソッド210によってデータをIStringクラス202に入力する方法を示す。

【0044】

フィールド	型	データ
baseString	Java文字列	Д а в и д К у м г и р
sortString	Java文字列	Д а в и д К у м г и р
altString	Java文字列	David Kumhyr
sourceLocale	Java文字列	R U
sourceLanguage	Java文字列	r u
targetLocale	Java文字列	U S
targetLanguage	Java文字列	e n

表III

【0045】以下に示す例では、ユーザによって入力される文字列は、baseString204およびsortString206の両方に挿入されるが、altString208に入力される文字列は、音訳エンジン220により、ロシア語のキリール文字から英語文字への音声マッピングのリソース・テーブルを使用して選択される。したがって、baseString204の発音どおりの表現は、キリール文字セットになじみのないユーザにとっての発音キーとしてaltString208に入力される。

【0046】ローマ字セットまたはローマ字セットに音声マッピングすることができる文字セットを使用しない言語の場合、データ入力メソッド210が、データをbaseString204、sortString206およびaltString208のフィールドに入力し、これは、入力メソッド・エディタ（IME）224から導出される。IME224は、カスタマイズされた入力メソッド・エディタであってもよい、Microsoft社（ワシントン州Redmond）から

市販されているWindows NTオペレーティング・システムのアジア版に統合された入力メソッド・エディタであってもよい。Windows NT入力メソッド・エディタを使用するならば、適当なデータをWindowsNT入力メソッド・エディタ内部データ記憶装置から抽出しなければならない。

【0047】表IVは、ローマ字セットをも、ローマ字セットに音声マッピングすることができる文字セットをも使用しない表意文字言語、たとえば日本語の場合にデータ入力メソッド210によってデータをIStringクラス202に入力する方法を示す。

【0048】

フィールド	型	データ
baseString	Java文字列	<Kanji>
sortString	Java文字列	ハヤシ
altString	Java文字列	hayashi
sourceLocale	Java文字列	J P
sourceLanguage	Java文字列	j a
targetLocale	Java文字列	U S
targetLanguage	Java文字列	e n

表IV

【0049】表意文字言語は、アルファベットをもたず、その代わり、単なる音ではなく概念およびオブジェクトに対応する記号（「表意文字」）の非常に大きな文字セットを有する。たとえば、1981年に採用された日本語の常用漢字表は1945個の記号を含む。通常のコンピュータ・キーボードは、この言語の1記号ごとに1個を割り当てられる数のキーを含むことはできず、そのため、入力は、キーストロークの組み合わせを発音どおりに使用して2種類の表意文字、ひらがなまたはカタカナのいずれかから文字を選択し、漢字記号の生成のための辞書検索して達成される。このプロセスは、上述したWindows NT入力メソッド・エディタで実現されている。

【0050】したがって、表意文字言語の場合、altString 208に入力されるデータは、所望の表意文字を構

```
/*
*****

```

```
*
*<P></P>
*
```

```
*

```

```
*<dt><b>Description:</b><dd>

```

```
*<p>Allocate a new IString. The baseString, sortString and

```

```
*altString are entered from the IME utilizing the default language

```

```
*and locale.</p>

```

```
*

```

```
*****

```

```
public IString(String base,
String sort,
String alt,
Locale src,
Locale tgt){
this.baseString=base;
this.sortString=sort;
this.altString=alt;
this.sourceLocale=src.getLocale();
this.sourceLanguage=src.getLanguage();
this.targetLocale=tgt.getLocale();
this.targetLanguage=tgt.getLanguage();
init();
}

```

【0053】baseString 204、sortString 206およびaltString 208のコンテンツは、IME 224から

*成するためにユーザによってタイプされるローマ字である。sortString 206に入力されるデータは、所望の表意文字を発音どおりにつづり、表意文字の中間的表現を提供する表意文字である。baseString 204に入力されるデータは、ユーザによって選択される最終的な表意文字である。上述したような非ローマ字の音訳と同様に、targetLanguage属性が英語以外の言語での発音どおりのつづりによる表意文字の構成をサポートするならば、非

ローマ字をaltString 208に入力することもできる。たとえば、IME 224が、ロシア語での発音どおりのつづりによる日本語の漢字の構成を可能にするならば、IStringオブジェクト 202は、日本語の漢字をbaseString 204に含み、ひらがなをsortString 206に含み、キリール文字をaltString 208に含むかもしれない。

【0051】IStringオブジェクト 202に入力するためのデータ入力メソッド 210によってbaseString 204、sortString 206およびaltString 208をIME 224から受けるのに適したコンストラクタは、以下のものであることができる。

【0052】

導出されたデータからそれぞれのフィールドに入力され、sourceLocale 212およびsourceLanguage 214の

コンテンツは、データがIStringオブジェクト202に入力されているホスト・システムによって指定されるデフォルト地域および言語属性から入力される。targetLocale216およびtargetLanguage218のコンテンツは、通常、ローマ字セットを使用する言語の地域/言語コード、たとえば「en...US」（英語-アメリカ合衆国）である。

【0054】文字列がIStringクラス・オブジェクト202に入力される時の言語にかかわらず、データ入力メソッド210によってbaseString204、sortString206およびaltString208それぞれに自動的に入力されるデータは、他のメソッドを使用して打ち消したり変更したりすることができる。IStringオブジェクト202

```
/******
```

```
*
* <P></P>
*
* <dt><b>Description:</b><dd>
* <p>Allocate a new IString. The baseString, sortString and
* altString are entered from the IME for specified target and source
* language and locale.</p>
*
```

```
*****
```

```
public IString(String base,
                String sort,
                String alt,
                String srLanguage,
                String srLocale,
                String tgtLanguage,
                String tgtLocale){
    this.baseString=base;
    this.sortString=sort;
    this.altString=alt;
    this.sourceLocale=srLocale;
    this.sourceLanguage=srLanguage;
    this.targetLocale=tgtLocale;
    this.targetLanguage=tgtLanguage;
    init();
}
```

【0057】このコンストラクタでは、baseString204、sortString206およびaltString208に入力される文字を選択するために使用されるソースおよび標的言語および地域を指定することができる。この後者のコンストラクタを使用して、ホスト・システム・デフォルト言語以外で、すなわち、IStringオブジェクト202のデータが別のシステムから受けられ、局所インスタンスが生成されるホスト・システムで、IStringオブジェクト202を生成することができる。

【0058】音訳エンジン220およびメッセージング・メソッド226は必ずしも図2に示すようにIString

*02のフィールドは、好ましくは、個々に独立して編集して、上述したようなsortStringフィールド206内の人工的なプロモーション、baseStringフィールド204中の誤選択された表意文字の交換またはaltString208内の発音どおりのつづりの訂正を可能にする。

【0055】上述したメソッドは、ソース言語および標的言語をホスト・システム・デフォルトからとる仮定したが、データはまた、コンストラクタを使用して、指定されたソース言語および標的言語のbaseString204、sortString206およびaltString208に入力してもよい。

【0056】

クラス202内で実現する必要はなく、IMEメソッド220を別々に実現する必要もないことを理解すべきである。代わりに、音訳エンジン220およびメッセージング・メソッド226を、必要に応じてIStringクラス202によって適切に構成および/または起動される別々のサブクラス内で実現し、IME224を、IStringクラス202内のメソッドとして実現してもよい。

【0059】音訳エンジン220およびIME224は、特定の地域および言語属性設定の下でIStringクラス202オブジェクトのための入力データを集めるためにデータ入力メソッド210によって要求されるだけで

23

ある。他の方法では、適当なコンストラクタを起動することによってデータをbaseString204、sortString206およびaltString208にプログラムの入力することもできる。IString202内でフィールドをプログラムの取得し、セットするために実行時にプログラムによって起動することができるメソッドは、以下を含む。

```

【0060】
/*****
 *
 * (P) (P)
 *
 * (dt) (b)Description: (b) (dd)
 * (p)Get the IString baseString. (p)
 *
 * @returns str String containing the base string
 *
 *****/
public String getBaseString(){
    return this.baseString;
}

```

【0061】 このメソッドは、IStringオブジェクト202のためのbaseString204のコンテンツを戻す。同様なメソッドが、sortString206およびaltString208のコンテンツを戻す。

【0062】

10

20

30

40

24

```

/*****
 *
 * (P) (P)
 *
 * (dt) (b)Description: (b) (dd)
 * (p)Get the IString sortString. (p)
 *
 * @returns str String containing the sort string
 *
 *****/
public String getSortString(){
    return this.sortString;
}

/*****
 *
 * (P) (P)
 *
 * (dt) (b)Description: (b) (dd)
 * (p)Get the IString altString. (p)
 *
 * @returns str String containing the alt string
 *
 *****/
public String getAltString(){
    return this.altString;
}

```

【0063】 このメソッドはまた、baseString204、sortString206およびaltString208の設定を含む。

```

【0064】
/*****
 *
 * (P) (P)
 *
 * (dt) (b)Description: (b) (dd)
 *
 * (p)Set the IString baseString. (p)
 *
 * @param str String containing the base string
 *
 *****/
public void setBaseString(String sBase){
    this.baseString=sBase;
}

```

【0065】

```

25
/*****
 *
 * (P) </P>
 *
 * (dt) (b)Description: </b> (dd)
 * (p) Set the IString sortString. </p>
 *
 * @param str String containing the sort string
 *
 *****/
10 public void setSortString(String sSrt){
    this.sortString=sSrt;
}

【0066】
/*****
 *
 * (P) </P>
 *
 * (dt) (b)Description: </b> (dd)
 * (p) Set the IString altString. </p>
20
 *
 * @param str String containing the alt string
 *
 *****/
public void setAltString(String sAlt){
    this.altString=sAlt;
}

【0067】 プログラムは、IStringオブジェクト20
2のためのbaseString204、sortString206および
30
/*****
 *
 * <P></P>
 *
 * (dt) <b>Description: </b> (dd)
 * (p) Get the locale of the IString data. </p>
 *
 * @returns loc Locale containing the locale of the data
 *
 *****/
public Locale getLocale(){
    Locale loc=new Locale(this.sourceLanguage, this.sourceLocale);
    return loc;
}

/*****
 *
 * <P></P>
 *
 * (dt) <b>Description: </b> (dd)
 * (p) Set the locale of the IString data. </p>

```

altString208の取得および設定に加えて、IStringオブジェクト202の表示地域または言語を取得または設定しなければならないかもしれない。したがって、プログラムがIStringデータの地域または言語属性を取得および/または設定することを許すための他のメソッドが提供される。
【0068】

```

27
*
**param loc Locale of the data
*
*****
public void setLocale(Locale loc){
    this.sourceLocale=loc.getLocale();
    this.sourceLanguage=loc.getLanguage();
}

/*****
*
*<P></P>
*
*<dt><b>Description:</b><dd>
*<p>Get the display language of the IString data.</p>
*
**returns Display language of the data
*
*****
public String getDisplayLanguage(){
    Locale loc=new Locale(this.sourceLanguage, this.sourceLocale);
    return loc.getDisplayLanguage();
}

/*****
*
*<P></P>
*
*<dt><b>Description:</b><dd>
*<p>Get the display locale of the IString data.</p>
*
**returns Display locale of the data
*
*****
public String getDisplayLocale(){
    if(this.sourceLanguage==null&&this.sourceLocale==null)
        return null;
    else{
        Locale loc=new Locale(this.sourceLanguage, this.sourceLocale);
        return loc.getDisplayLocale();
    }
}

```

【0069】これらのメソッドは利用可能であるが、IStringクラス202は、好ましくは、プログラマ/ユーザがIStringクラス202のために実現されるメソッドに関して何も知る必要がないような「ブラック・ボックス」動作を示す。IStringクラス202は、baseString204に関する余計な情報をカプセル化し、文字のある文字セットから別の文字セットに変換するためのいくつかのメソッドを含むデータ型として見えるだけである。baseString204に加えて、またはその代わりにsortStringフィールド206またはaltStringフィールド20

8が編集のため、または表示だけのためにユーザに露呈されなければならない。特殊な場合に備えて、別個のセットの制御を提供してもよい。

【0070】本発明では、IStringクラス202は、異なる言語を使用するシステム間で人間言語データを効果的に転送するために使用される。baseString204のコンテンツは、IStringオブジェクト202を生み出すシステムのデフォルト言語にある文字列の固有の表現を提供する。しかし、各システムが、異なる人間言語で稼働する他のシステムとでデータの交換に参与する場合、IS

29

tringオブジェクト202のtargetLocale属性216およびtargetLanguage218属性は、好ましくは、共通の値に設定される(たとえばtargetLocale=「U.S」、targetLanguage=「en」)。したがって、altString208のコンテンツは、言語間で共通する文字列の表現を含む。オブジェクトを受けるシステムのデフォルト言語がbaseString204のコンテンツの言語から異なるシステムでは、IStringクラス・オブジェクト202は、altString208のコンテンツを、表示または処理すべき文字列として提示するよう、自動的に切り換えることがで

きる。
【0071】図3を参照すると、本発明の実施態様にしたがって、個々のマルチフィールド文字列クラス・オブジェクトまたはマルチフィールド文字列クラス・オブジェクトの離散的セットの表示および編集を制御するための機構が示されている。すべてのJava AWT (Abstract Windowing Toolkit) 制御を動作可能にしてIString情報を、表示するために、文字列マネージャ・クラス、IStringControl302およびIStringコントローラ・インタフェース312が定義されている。IStringコントローラ・インタフェース312は、IStringオブジェクト304に対するすべての表示要求をインタセプトして、そのような表示要求にตอบสนองして、現表示フィールドのコンテンツを制御にパスする。したがって、IStringコントローラ・インタフェース312は、専用の表示制御として、レガシー制御がIStringオブジェクト304のコンテンツを表示することを可能にする。

【0072】IStringControl302のコンストラクタは、そのパラメータとして、IStringクラス304およびJFC (Java Foundation Class) 制御306、メニュー項目308またはsetText()メソッドを有する他の制御310をとる。IStringControl302は、どのIStringフィールド (baseString、sortStringまたはaltString) を、それが管理する制御の中で見せるかを管理し、IStringオブジェクト304内で文字列を入力または変更するためのメソッドを実現する責任を負う。IStringControl302は、制御のsetText()メソッドに対するすべての呼び出しに介入して、制御を使用しながらIStringオブジェクト304のフィールドに文字列を入力するために必要な処理を実行する。

【0073】ユーザ・インタフェース対話内の文字列制御のインスタンスごとに、JFC制御にsetText()メソッドを呼び出す代わりに、IStringコントローラ・インタフェース312およびJFC制御が、JFC制御を管理するためのIStringControl302とともに生成される。これは、多数の単一値文字列ベースの制御(たとえばJLabelまたはJTextField)を含む対話またはフレームがIStringを見せることを可能にする。さらに、IStringControl302は、JFC制御をそのコンストラクタ・パラメータの1種としてとる代わりに、jav

30

a.awt.Componentをとり、内容をを使用してsetText()メソッドを渡し、IStringサポートを潜在的に偏在させることもできる。

【0074】IStringコントローラ (IStringControlActions) インタフェース312はまた、IStringControl302を構成し、照会するためのメソッドによって定義される。そのようなメソッドは、以下を含む。

【0075】
public void setText(IString text),
public IString getText(),
public void setVisibleField(int field),
public int getVisibleField(),
setPopupEnabled(boolean enabled),および
isPopupEnabled().

【0076】getText()およびgetVisibleField()メソッドは、コントローラ・インタフェース312内で無効化されず、コントローラ・インタフェース312が、現表示フィールド (「visibleField」) のコンテンツをレガシー制御、たとえばJava AWT制御、たとえばJLabel制御306、JFC制御、たとえばメニュー項目308またはgetText()メソッドを含む他の制御310にパスすることを許す。

【0077】専用のIStringControl302が、レガシー制御内のsetText()メソッドに対する呼び出しに介入するはすであるため、コントローラ・インタフェース312内のgetText()およびsetVisibleField()メソッドは無効化される。上述したメソッド中のパラメータ「フィールド」は、IStringControl.BASE、IStringControl.SORTまたはIStringControl.ALTOのいずれかである。上述した最後の二つのメソッドが、ポップアップ表示をIStringフィールド・コンテンツとともに示すかどうかを決定する。

【0078】さらに、IStringControlOptionsメソッドを実現する、ButtonGroupに類似し、例ではIStringControlGroupと呼ばれるグルーピング・クラス314を定義することができる。IStringControl302のインスタンス316をグルーピング・クラス314に加えると、すべての制御を一度に構成することができる文脈、たとえば対話または全アプリケーションを生成することができる。図3に示し、上述した構造では、アプリケーションおよびプログラマでさえ、アプリケーション内のIStringデータを使用するために実行すべき余計な作業がほとんどないが、所望により、特異な操作のためのメソッドのより豊富なセットを使用してもよい。

【0079】コントローラ・インタフェース312はまた、メッセージングを介してIStringデータの表示を変更するための機構を提供する。言語および/または地域コードを含むメッセージ、たとえば言語変更メッセージ、地域変更メッセージまたは表示変更メッセージに関して、IStringデータ・オブジェクト304、制御30

31

2、306、308および310ならびにグループ・コントロール314は、すべてリスナ(listener)として、コントロール・インタフェース312に登録することができる。言語および地域変更メッセージは、基礎にあるシステム内で言語および地域属性設定を変更することができるが、表示変更メッセージは、別に、基礎にあるシステムの言語および地域属性を変更することなくIStringデータ表示フィールドを選択するために使用される言語および/または地域を変更する(別々に、または含めて)ことができるだけである。

【0080】言語、地域および表示変更メッセージは、ユーザがコントロール・インタフェース312との対話を介して起動することもできるし、アプリケーションが適当なメッセージを送ることによって起動することもできる。新たな言語および/または地域が選択されると、動作可能にされたすべての加入者は、人間の言語をロードし直し、地域特定のデータをフォーマットし直すことによってメッセージに対応することができる。言語、地域および表示変更メッセージは、図2に示すIStringオブジェクト202内で1個以上のメッセージング・メソッド224を呼び出すことができる。

【0081】言語変更メッセージを受けると、IStringオブジェクト302は、それらのデータに対して論理的に演算を加える。いくつかのIStringオブジェクト302は、言語の変更に対して挙動変化を示さず、地域の変化に対して挙動変化を示すかもしれない。たとえば、数値データ表示は、選択された地域(たとえば米国からヨーロッパ)に依存してその提示フォーマットを変更するが、数字は変更しないであろう。他のオブジェクトは、提示フォーマットならびにテキストの月名および曜日を、表示変更するテキスト日付表示におけるように、言語および地域の両方に関して挙動変化を有する。

【0082】図4および5を参照すると、本発明の好ましい実施態様にしたがって、マルチフィールド文字列クラス・オブジェクトの表示を制御する方法のユーザ・インタフェース図が示されている。図4は、オブジェクト名が含まれるユーザ・インタフェースを示す。このユーザ・インタフェースの中では、マルチフィールド文字列オブジェクトが、例示的な実施態様では、漢字記号を発音とおりにつづるカタカナ文字をbaseString(図示せず)とともに含むsortStringフィールドである第一のフィールドのコンテンツとともに表示されている。

【0083】図3に示し、上述した制御は、マルチフィールド文字列オブジェクトの表示を制御するために使用することができる。図5は、オブジェクト名の表示フィールドがaltStringフィールドのコンテンツに変更された同じオブジェクトを表示するユーザ・インタフェースを示す。

【0084】次に図6および7を参照すると、本発明の好ましい実施態様にしたがって、マルチフィールド文字

32

列オブジェクトの現表示フィールドを選択または編集するためのユーザ・インタフェース制御が示されている。図6および7は、図2および3と関連させて参照するためのものである。制御502は、好ましくは、IStringオブジェクト内のどのフィールドのコンテンツをも要求することができる専用のIString制御302である。

【0085】制御502は、IStringオブジェクト202または304の現表示フィールドのコンテンツを表示するための表示区域504と、表示すべき異なるフィールドの選択を起動するための選択制御506とを含む。IStringオブジェクト202または304の現在選択されている表示フィールドのコンテンツは、表示のためのsetVisibleFieldメソッドによって検索することができる。IStringオブジェクト202または304内のvisibleFieldパラメータが、どのフィールド、baseString204、sortString206またはaltString208が、現在表示に選択されているかを識別する。visibleFieldパラメータに基づき、setVisibleFieldメソッドは、「フィールド」パラメータがIString.BASE、IString.SORTまたはIString.ALTであるIStringオブジェクト202または304に対してgetText(field)プロセスを効果的に実行して、現在表示に選択されているフィールド内の文字列を検索する。

【0086】選択制御506は、選択制御506をクリックし、カーソルが選択制御506の上に来るまでポインティング・デバイス进行操作したのち、ポインティング・デバイスまたはポインティング・デバイス上の特定のボタンを起動することによって起動される従来の下向き矢印であってもよい。起動されると、選択制御506は、IStringオブジェクト202または304の三つのフィールドすべてをbaseString204、sortString206およびaltString208の順で含むドロップダウン・ボックス508を生成する。三つのフィールドすべてのコンテンツは、IString.BASE、IString.SORTおよびIString.ALTに対して複数のgetText()メソッドを実行することによって検索することができる。すると、三つの文字列を、公知の技術にしたがってドロップダウン・ボックス508内に順序付きの一覧として表示することができる。

【0087】従来のプルダウン・メニューおよびドロップダウン・ボックスと同様に、カーソルが現在位置するドロップダウン・ボックス508内のエントリがパー510によって強調表示される。ユーザは、ドロップダウン・ボックス508内の所望のエントリをクリックすることによって異なるフィールドを表示に選択することができる。すると、制御502がsetVisibleTextメソッドを実行してvisibleTextパラメータを適宜にIString.BASE、IString.SORTまたはIString.ALTに変更したのち、新たに選択された現表示フィールドのコンテンツを表示区域504にロードし直す。

【0088】ユーザは、編集すべきフィールドが表示されている間に表示区域504をダブルクリックすることにより、残りのフィールドのコンテンツを変更することなく、現在選択された表示フィールドのコンテンツを編集することができる。いかなる文字を含むことができるユーザ指定文字列がgetText()メソッドによって現在表示されているフィールドに記憶される。現在表示されていないフィールドのコンテンツを編集するためには、ユーザは、選択制御506を使用して編集すべきフィールドを現表示フィールドとして選択する。そして、表示区域504内のフィールドを編集する。

【0089】制御502は、状態バー内で適切に実現することもできるし、図7に示すように、対話ボックス512の中に他の制御とともに統合することもできる。図示するように、他の制御は、たとえば、選択制御506が動作可能であるか、動作不能であるか、また、表示区域504のコンテンツが読取り専用であるか、編集可能であるかを制御するラジオ・ボタン514を含むことができる。このように、選択制御506および表示区域504の対応するパラメータを単独で、または組み合わせで使用しながら、制御502は、ユーザが、指定のIStringオブジェクトの現表示フィールドを変更したり、指定のIStringオブジェクトでフィールドのコンテンツを編集したり、指定のIStringオブジェクト内の指定のフィールドのコンテンツを編集したりすることを防ぐことができる。

【0090】次に図8を参照すると、本発明の好ましい実施態様にしたがって、マルチフィールド文字列内のフィールドを選択的に表示または編集するためのユーザ制御の動作の高レベル流れ図が示されている。図示するプロセスは、図6および7に示す制御502が動作するためのプロセスである。プロセスはステップ602で始まり、ユーザ制御502が起動されることによってIStringオブジェクトを表示したとき、ステップ604に進んで、どのフィールドが現表示フィールドであるかを判定し、そのフィールドのコンテンツを制御内に表示する。

【0091】次にプロセスはステップ606に進み、ユーザが選択制御506を起動するなどによって表示フィールドの変更が起動されたかどうかを判定する。起動されたならば、プロセスはステップ608に進み、IStringオブジェクトのすべてのフィールドをドロップダウン・ボックス508中に順に表示し、新たなフィールドが現表示フィールドとして選択されるならば、表示ウィンドウ504のコンテンツを変更する。ひとたびユーザが表示のためのフィールドを選択したならば、ドロップダウン・ボックス508は除かれる。

【0092】現表示フィールドの変更が起動されないのならば、プロセスはステップ610に進み、ユーザが表示ウィンドウ504をダブルクリックするなどによって現表示フィールドのコンテンツの編集が起動されたかど

うかを判定する。起動されたならば、プロセスはステップ612に進み、IStringのソース言語および標的言語ならびに編集されるフィールドに依存して、簡単なメディア、音訳エンジンまたはIMEを使用して、現表示フィールドのコンテンツを編集する。

【0093】ステップ608または612のいずれから、プロセスは次に、ステップ614に進み、制御502によるIStringオブジェクトの表示が終了したかどうかを判定する。終了していないならば、プロセスはステップ606に戻り、表示フィールドの変更または編集のさらなるユーザ起動を待つ。しかし、終了したならば、プロセスはステップ616に進み、制御502を使用してIStringオブジェクトの別の表示が起動されるまでアイドル状態になる。

【0094】本発明のユーザ制御は、ユーザが、IStringオブジェクトの異なるフィールドの間でビューを切り換えて文字列の異なる表現を見ることができ、IStringクラスの利点を導出することを可能にする。現表示フィールド内に記憶された表現に基づいて文字列を認識、理解または発音することができないユーザは、別の表現、たとえば文字列の発音どおりのつづりに切り換えることができる。これが、ある言語で入力された文字列を、その文字列が入力されたときの音質に近似的でないユーザが認識、理解または発音することを可能にする。

【0095】本発明を完全に機能的なデータ処理システムおよび/またはネットワークに関連して説明したが、当業者が本発明の機構を多様な形態のコンピュータ使用可能な命令媒体の形態で分散させられることを認識し、本発明がその分散を実際に実施するために使用される信号搬送媒体の特定のタイプにかかわらず等しく適用されることを理解することが重要である。コンピュータ使用可能な媒体の例は、不揮発性のハードコード化タイプ媒体、たとえば読み取り専用メモリ（ROM）または消去可能な電気的にプログラム可能な読み取り専用メモリ（EPROM）、記録可能なタイプの媒体、たとえばフロッピー・ディスク、ハードディスク、ドライブおよびCD-ROMならびに伝送タイプ媒体、たとえばデジタルおよびアナログ通信リンクを含む。

【0096】好ましい実施態様を参照しながら本発明を具体的に示し、説明したが、本発明の本質および範囲を逸することなく、形態および詳細における種々の変更を加えてもよいことが当業者によって理解されるであろう。

【0097】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) マルチフィールド文字列の表示を変更する方法であって、ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタファースを提供するステップと、前記マルチフィールド文字列オブジェクトの現

35

表示フィールドのコンテンツを前記ユーザ制御中に表示するステップと、前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するステップと、前記選択制御のユーザ起動に応答して、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するステップと、を含むことを特徴とする方法。

(2) ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供する前記ステップが、前記マルチフィールド文字列オブジェクトの第一のフィールド内の第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するステップと、前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するステップと、をさらに含む上記(1)記載の方法。

(3) 前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示する前記ステップが、前記マルチフィールド文字列オブジェクトのどのフィールドが前記現表示フィールドであるかを識別するステップと、識別した前記フィールドから文字列を検索するステップと、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するステップと、をさらに含む上記(1)記載の方法。

(4) 前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択に応答して、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するステップをさらに含む上記(1)記載の方法。

(5) 前記新たな現表示フィールドから文字列を検索するステップと、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するステップとをさらに含む上記(4)記載の方法。

(6) 前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するステップと、前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するステップと、前記選択制御のユーザ起動に応答して、前記現表示フィールドに入力するための新たな文字列を受けるステップと、前記新たな文字列を前記マルチフィールド文字列オブジェクト内の前記現表示フィールド中に記憶するステップとをさらに含む上記(1)記載の方法。

(7) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示する前記ステップが、前記文字列が入力された

36

きの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するステップと、前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するステップとをさらに含む上記(1)記載の方法。

(8) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示する前記ステップが、前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するステップと、前記文字列を発音どおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するステップと、前記文字列を発音どおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するステップとをさらに含む上記(1)記載の方法。

(9) マルチフィールド文字列の表示を変更するためのシステムであって、ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための手段と、前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための手段と、前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するための手段と、前記選択制御のユーザ起動に応答して、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための手段と、を含むことを特徴とするシステム。

(10) ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための前記手段が、前記マルチフィールド文字列オブジェクトの第一のフィールド内で第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するための手段と、前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するための手段と、をさらに含む上記(9)記載のシステム。

(11) 前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための前記手段が、前記マルチフィールド文字列オブ

プロジェクトのどのフィールドが前記現表示フィールドであるかを識別するための手段と、識別した前記フィールドから文字列を検索するための手段と、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための手段と、をさらに含む上記(9)記載のシステム。

(12) 前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択に応答して、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するための手段をさらに含む上記(9)記載のシステム。

(13) 前記新たな現表示フィールドから文字列を検索するための手段と、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための手段とをさらに含む上記(12)記載のシステム。

(14) 前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するための手段と、前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するための手段と、前記選択制御のユーザ起動に応答して、前記現表示フィールドに入力するための新たな文字列を受け取るための手段と、前記新たな文字列を前記マルチフィールド文字列オブジェクト内の前記現表示フィールド中に記憶するための手段とをさらに含む上記(9)記載のシステム。

(15) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記手段が、前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための手段と、前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための手段とをさらに含む上記(9)記載のシステム。

(16) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記手段が、前記文字列が入力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための手段と、前記文字列を発音とおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための手段と、前記文字列を発音とおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するための手段とをさらに含む上記(9)記載のシステム。

(17) マルチフィールド文字列の表示を変更するためのコンピュータ使用可能な媒体内のコンピュータ・プロ

グラム製品であって、ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための命令と、前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための命令と、前記マルチフィールド文字列オブジェクトの現表示フィールドを選択的に変更するための選択制御を提供するための命令と、前記選択制御のユーザ起動に応答して、前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための命令と、を含むことを特徴とするコンピュータ・プログラム製品。

(18) ユーザ制御と、それぞれが文字列の異なる表現を含む別個のフィールドを有するマルチフィールド文字列オブジェクトとの間にインタフェースを提供するための前記命令が、前記マルチフィールド文字列オブジェクトの第一のフィールド内で第一の人間言語によって前記文字列を前記第一の人間言語で表すために使用される第一の文字セットからの文字を含む第一の文字列を記憶するための命令と、前記マルチフィールド文字列オブジェクトの第二のフィールド内で第二の人間言語によって前記文字列を前記第二の人間言語で表すために使用される第二の文字セットからの文字を含む第二の文字列を記憶するための命令と、をさらに含む上記(17)記載のコンピュータ・プログラム製品。

(19) 前記マルチフィールド文字列オブジェクトの現表示フィールドのコンテンツを前記ユーザ制御中に表示するための前記命令が、前記マルチフィールド文字列オブジェクトのどのフィールドが前記現表示フィールドであるかを識別するための命令と、識別した前記フィールドから文字列を検索するための命令と、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための命令と、をさらに含む上記(17)記載のコンピュータ・プログラム製品。

(20) 前記一覧からの前記マルチフィールド文字列オブジェクト内のフィールドの新たな現表示フィールドとしてのユーザ選択に応答して、前記マルチフィールド文字列オブジェクト内の前記現表示フィールドを前記新たな現表示フィールドに変更するための命令をさらに含む上記(17)記載のコンピュータ・プログラム製品。

(21) 前記新たな現表示フィールドから文字列を検索するための命令と、検索した前記文字列を前記ユーザ制御内の表示区域中に表示するための命令とをさらに含む上記(20)記載のコンピュータ・プログラム製品。

(22) 前記現表示フィールドからの文字列を前記ユーザ制御内の表示区域中に表示するための命令と、前記現表示フィールドのコンテンツを選択的に編集するための編集制御を前記ユーザ制御内に提供するための命令と、前記選択制御のユーザ起動に応答して、前記現表示フィールドに入力するための新たな文字列を受け取るための命

行と、前記新たな文字列を前記マルチフィールド文字列オブジェクト内の前記現表示フィールド中に記憶するための命令とをさらに含む上記(17)記載のコンピュータ・プログラム製品。

(23) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記命令が、前記文字列が人力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための命令と、前記第一の文字列内の文字に音声マッピングする文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための命令とをさらに含む上記(17)記載のコンピュータ・プログラム製品。

(24) 前記マルチフィールド文字列オブジェクト内の全フィールドのコンテンツをユーザ選択のための一覧として表示するための前記命令が、前記文字列が人力されたときの第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第一のフィールドからの第一の文字列を表示するための命令と、前記文字列を発音どおりにつづるために前記第一の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第二のフィールドからの第二の文字列を表示するための命令と、前記文字列を発音どおりにつづる第二の人間言語によって使用される文字を含む、前記マルチフィールド文字列オブジェクト内の第三のフィールドからの第三の文字列を表示するための命令とをさらに含む上記(17)記載のコンピュータ・プログラム製品。

【図面の簡単な説明】

【図1】本発明の好ましい実施態様を実現することができるとするデータ処理システムを示す図である。

【図2】本発明の好ましい実施態様にしたがって、ユーザ・インタフェースに代替表示言語選択を提供するために使用されるマルチフィールド文字列クラスを示す図である。

【図3】本発明の好ましい実施態様にしたがって、個々の

のマルチフィールド文字列クラス・オブジェクトまたはマルチフィールド文字列クラス・オブジェクトの離散的セットの表示および編集を制御するための機構を示す図である。

【図4】本発明の好ましい実施態様にしたがって、マルチフィールド文字列クラス・オブジェクトの表示を制御する方法を示すユーザ・インタフェース図である。

【図5】本発明の好ましい実施態様にしたがって、マルチフィールド文字列クラス・オブジェクトの表示を制御する方法を示すユーザ・インタフェース図である。

【図6】本発明の好ましい実施態様にしたがって、マルチフィールド文字列クラス・オブジェクトの現表示フィールドを選択または編集するためのユーザ・インタフェース制御を示す図である。

【図7】本発明の好ましい実施態様にしたがって、マルチフィールド文字列クラス・オブジェクトの現表示フィールドを選択または編集するためのユーザ・インタフェース制御を示す図である。

【図8】本発明の好ましい実施態様にしたがって、マルチフィールド文字列内のフィールドを選択的に表示または編集するためのユーザ制御の動作を示す高レベル流れ図である。

【図9】多数の意味または発音を有する一つの表意文字を示す図である。

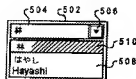
【図10】多数の意味または発音を有するもう一つの表意文字を示す図である。

【図11】多数の意味または発音を有するさらに別の表意文字を示す図である。

【符号の説明】

202 Istringクラス
204、206、208、212、214、216、218 フィールド
210 データ入力メソッド
220 音訳エンジン
222 Javaリソース・ファイル
224 入力メソッド・エディタ
226 メッセージング・メソッド

【図6】



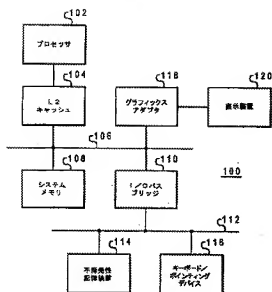
【図9】

西 林 菅 野

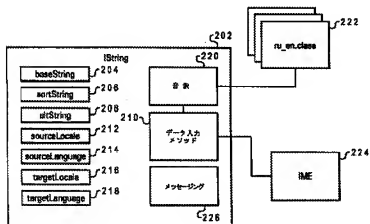
【図10】

【図11】

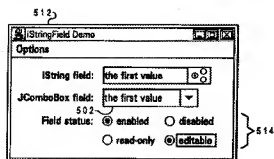
【図1】

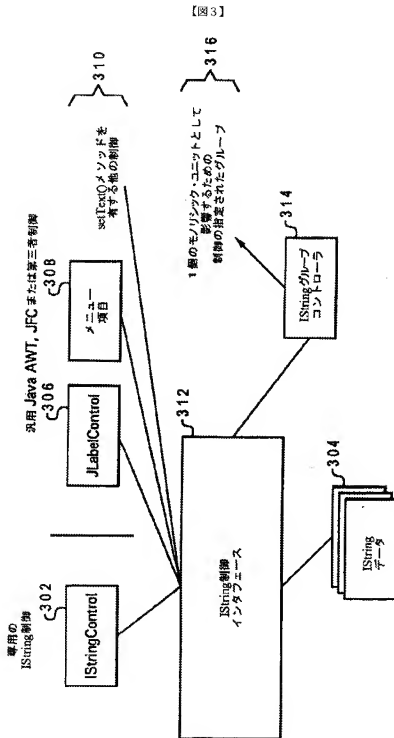


【図2】

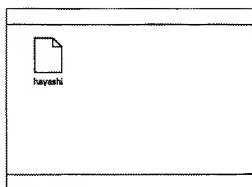


【図7】

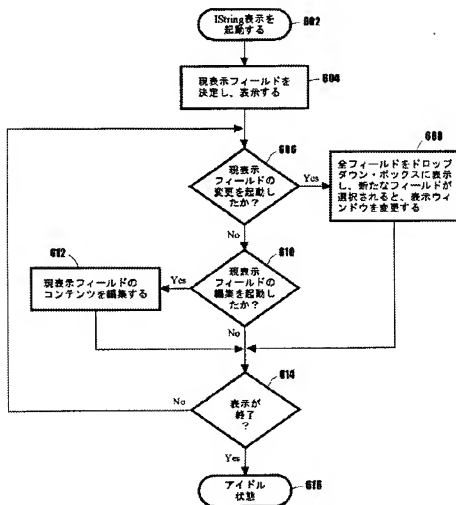




【図5】



【図8】



フロントページの続き

(72)発明者 デヴィット・ジェームス・ヘサーリントン
アメリカ合衆国78759 テキサス州、オ
ースティン、レイン・クリーク・パーク
ウェイ 7397

(72)発明者 デヴィット・ブルース・クマイア
アメリカ合衆国27526 ノース・カロライ
ナ州、ファガIVERアリナ、ウェス
ト・アカデミー・ストリート 604